# A Hybrid Ensemble End-to-End Neural Network for Accurate Protein-Protein Interactions Prediction

Jie Yang, Xijie Lan, Guoyin Wang, *Senior Member, IEEE*, Zhong Chen, Yuwen Chen, and Di Wu, *Member, IEEE*

***Abstract*—Protein-protein interactions (PPIs) are fundamental to understanding cellular mechanisms, signaling networks, disease pathways, and drug development. Over the years, numerous computational models with artificial intelligence (AI) have been developed to predict PPIs. However, these models mostly face significant challenges, such as fragmented feature extraction pipelines, inability to capture complex global relationships among proteins, and reliance on handcrafted features. These challenges often limit their prediction accuracy. To address these issues, the Knowledge Graph Fused Graph Neural Network (KGF-GNN) was proposed, offering an end-to-end learning approach that integrates Protein Associated Network (PAN) with observed PPI data. While KGF-GNN achieves notable performance improvements, it focuses primarily on local topological features extracted by Graph Neural Networks (GNNs), potentially overlooking critical global patterns. Moreover, its feature fusion process lacks the flexibility to effectively combine diverse biological information. To overcome these shortcomings, this paper introduces a Hybrid Ensemble End-to-End Neural Network (HEENN), which incorporates three key innovations: (1) *Local Feature Extraction via Graph Attention Network (GAT):* HEENN employs GAT to enable more precise extraction of local topological and semantic features, allowing the model to focus on the most relevant interactions and relationships within the data. (2) *Global Feature Extraction via AutoEncoder:* By leveraging an AutoEncoder framework, HEENN captures comprehensive global features from PANs and PPI datasets, complementing the GAT's local features to produce richer protein representations. (3) *Attention-Enhanced Feature Fusion:* An attention mechanism is employed during feature fusion to ensure an adaptive and effective integration of local and global features. Extensive experiments on real-world PPI datasets demonstrate that HEENN significantly outperforms KGF-GNN and other state-of-the-art models, achieving superior accuracy in PPI prediction. These advancements underscore the potential of HEENN in AI-driven bioinformatics research, which offers new opportunities for biological discovery and therapeutic innovation.**

***Index Terms*—Protein-protein interactions prediction, graph attention network, knowledge graph, PPI network, AutoEncoder.**

Received 28 May 2025; revised 23 July 2025; accepted 24 July 2025. Date of publication 29 July 2025; date of current version 10 December 2025. This work was supported in part by the National Science Foundation of China under Grant 62221005, Grant 62466063, Grant 62176070, and Grant 62371438, in part by the Guizhou Provincial Department of Education Colleges and Universities Science and Technology Innovation Team under Grant QJJ[2023]084, in part by the New Chongqing Youth Innovation Talent Project under Grant CSTB2024NSCQ-QCXMX0035, in part by Leading Project of "Xinzhi Qianyan" by Guizhou Provincial Association for Science and Technology under Grant 2025XZQYXM-02, in part by the Chongqing Municipal Natural Science Foundation under Grant CSTB2024NSCQ-MSX1043, and in part by the Guizhou Provincial Key Laboratory of Evolutionary Artificial Intelligence under Grant QJJ[2022] 059. *(Corresponding author: Yuwen Chen.)*

Jie Yang is with the School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China, and also with the School of Physics and Electronic Science, Zunyi Normal University, Zunyi 563002, China (e-mail: yj530966074@foxmail.com).

Xijie Lan is with the School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: lanxi0923@gmail.com).

Guoyin Wang is with the National Center for Applied Mathematics in Chongqing, Chongqing Normal University, Chongqing 401331, China (e-mail: wanggy@cqnu.edu.cn).

Zhong Chen is with the School of Computing, Southern Illinois University, Carbondale, IL 62901 USA (e-mail: zhong.chen@cs.siu.edu).

Yuwen Chen is with the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China (e-mail: chenyuwen@cigit.ac.cn).

Di Wu is with the College of Computer and Information Science, Southwest University, Chongqing 400715, China (e-mail: wudi.cigit@gmail.com).

The code and dataset for the HEENN model: https://github.com/lxj923/HEENN_code.

This article has supplementary downloadable material available at https://doi.org/10.1109/TCBBIO.2025.3593469, provided by the authors.

Digital Object Identifier 10.1109/TCBBIO.2025.3593469

## I. Introduction

PROTEIN are organic macromolecules made up of amino acids, which are essential components of cells and sus-tain life activities. They play an important role in biology by linking various important physiological activities ofcells to PPIs [1], thereby facilitating a range of life activities such as apoptosis and immune response. Protein-protein interactions (PPIs) play an important role in cell life activities such as transcriptional regulation, signal transduction and drug signal transduction. Studying PPIs helps us understand cell signaling, disease mechanisms, and drug development [2], [3]. Many drugs exert their effects by targeting and interrupting specific protein interactions. Consequently, a deeper understanding of these interactions and their underlying mechanisms is essential for designing more efficient and targeted treatment strategies.

In recent years, high-throughput lab techniques like yeast two-hybrid, mass spectrometry, and protein microarrays have been used to study PPIs. However, all of these are based on biological and chemical experiments, which require substantial human, financial and time resources. To address this, computational methods based on artificial intelligence (AI) have become popular in bioinformatics [4], [5]. AI techniques provide powerful tools for processing large-scale datasets, modeling complex interactions, and accelerating the identification of novel PPIs. This capability enhances our understanding of cellular processes and supports advancements in drug research and development [6].

At present, several recent models have begun to incorporate both local and global features to improve PPI prediction. DSSGNN-PPI employ dual-graph or teacher–student frameworks to jointly model sequence and network-level information under limited supervision [7]. KSGPPI and MFCPPI integrate protein language embeddings, structural embeddings, and network topology features, with contrastive learning strategies to enhance prediction generalization and feature synergy across modalities [8], [9]. MGPPI and ProAffinity-GNN explore structure-informed learning through multi-scale GCNs and binding affinity regression, improving interpretability [10], [11]. PROBind aggregates residue-level predictions from multiple sources, showcasing the potential of fine-grained feature integration [12]. Despite these advances, most methods still rely on multi-stage processing or rigid fusion strategies. KGF-GNN, for instance, focuses on local topological features but lacks flexibility in global context modeling [13]. In contrast, our proposed HEENN framework adopts a unified, end-to-end strategy that seamlessly fuses local and global representations through attention-based mechanisms, achieving more accurate and adaptable PPI prediction.

To further improve PPI prediction and fix problems in feature fusion, we propose a Hybrid End-to-End Neural Network (HEENN) model with three key innovations. First, we extend the GNNs by integrating an attention mechanism, transforming them into Graph Attention Networks (GATs). This enhancement enables the model to prioritize critical relationships between proteins and adaptively refine feature extraction from complex biological networks. Second, we use an AutoEncoder in both the Protein Associated Network (PAN) and the PPI network to extract global features. The AutoEncoder framework is used to extract global features from each network. It complements the local topological and semantic features extracted by the respective GATs. This ensures a more comprehensive representation of protein interactions. Third, we use an attention-based multi-layer perceptron to combine global and local features. This makes feature fusion more effective and flexible. These advancements enable HEENN to achieve superior performance compared to existing models, addressing the limitations of KGF-GNN and providing new insights into protein interaction mechanisms.

The HEENN model uses a fully end-to-end architecture. All submodules are integrated without the need for manual feature engineering or step-by-step processing. The model handles data preprocessing and feature extraction automatically. During training, it uses gradient descent with backpropagation to update parameters across the entire network. Each module learns and adjusts its parameters dynamically. This allows the model to continuously improve its overall performance. The main contributions of this work are as follows:

- Two knowledge graphs, the Protein Associated Network (PAN) and PPI Network, are utilized to capture the intricate relationships between proteins, drugs, diseases, ribonucleic acids, protein structures, and other biological components. This dual-graph approach ensures that both topological and semantic features are captured. It offers a complete view of protein interactions and their links to various biological factors.
- A novel end-to-end model, HEENN, is introduced to overcome the limitations of traditional models. By integrating the AutoEncoder framework into both the PAN and PPI networks, the model enhances the extraction of global features. At the same time, attention mechanisms are incorporated within the GNNs and the feature fusion module. This setup enables HEENN to adaptively learn and combine both global and local features for more accurate PPI prediction.
- The effectiveness of HEENN is demonstrated through extensive experiments on real-world PAN and PPI datasets. We compare HEENN with classic and state-of-the-art models. The results show that our model performs better and is more robust. It predicts PPIs more accurately and offers deeper insights into protein interaction mechanisms.

Overall, these contributions highlight the significance and novelty of our approach, which integrates PAN and PPI Networks within the HEENN model to improve PPI prediction. By adding an attention mechanism to GNN to form GAT and integrating the AutoEncoder framework for feature extraction, our method can effectively capture local and global features. As a result, our method offers new insights into complex protein interactions. It also improves prediction accuracy and supports the use of AI in bioinformatics.

## II. Related Work

Predicting protein-protein interactions (PPIs) is a key challenge in computational biology. Over the years, researchers have proposed various methods to address this task. Early machine learning models, such as SVMs, made initial progress in PPI prediction. However, they relied heavily on handcrafted features, which limited their performance.

One of the first deep learning-based models, DeepPPI (2017), utilized deep neural networks to predict PPIs from protein sequence data. DeepPPI showed considerable improvement over traditional machine learning models by automatically learning complex feature representations from raw protein sequences. However, DeepPPI did not fully capture interactions between protein pairs. This limited its ability to process complex biological data [14]. PIPR (2019) introduced a more advanced architecture with a deep residual recurrent convolutional network. This model was designed to better capture local and sequential features from protein sequences, making it particularly effective for predicting homodimeric and heterodimeric interactions [15]. To address the need for structural data, DeepTrio (2022) introduced a new framework. It used masked multiscale CNNs to capture contextual information at different scales from protein sequences [16]. DeepTrio showed the benefits of using multiscale features. However, its performance was lower than other models on some datasets. For example, on the S. cerevisiae core dataset, it was outperformed by PIPR and DeepFE-PPI [17]. The S-VGAE (2020) model marked a shift towards graph-based models by utilizing a signed variational graph autoencoder (S-VGAE) to integrate protein sequences and graph structures. S-VGAE is able to capture both sequence and network features, but its focus on graph structure alone left gaps in its ability

to handle long-range dependencies in protein sequences. Additionally, it was not well-equipped to process the complex relationships between proteins, diseases, and other biological entities [18].

Further advances in graph-based models resulted in the development of Struct2Graph (2020), which employed graph convolutional networks (GCNs) to process 3D protein structural data [19]. This model significantly improved the accuracy of PPI predictions by utilizing structural information to capture protein interactions. However, Struct2Graph depended on the availability of 3D structures. This limited its usefulness, since not all proteins have clearly defined 3D structures. The SPNet (2021) introduced the Siamese pyramid network, which predicted binding probabilities between two proteins based on their amino acid sequences. This model incorporated an attention mechanism and multilevel feature structures to enhance prediction performance. However, SPNet was computationally expensive. It also relied heavily on sequence data, which limited its ability to deal with incomplete datasets [20]. BiLSTM-RF (2021) was another notable approach with combining bidirectional LSTM networks with random forest classifiers. It extracted and integrated features from protein pair sequences to predict PPIs. Although the model showed promise, it was computationally expensive and had long training times, making it less practical for large-scale applications [21]. In addition, D-SCRIPT (2021) applied a pre-trained language model to extract structural features from protein sequences. This method worked well for predicting PPIs across different species. However, it did not perform as well for within-species predictions due to limited training data. This highlighted the difficulty of building models that generalize across species [22].

Although these models improved PPI prediction, most relied on either sequence-based or structure-based information. Few attempted to integrate both types of data effectively. This narrowed their scope and limited their ability to capture the full complexity of protein interactions in biological networks. Unlike earlier models that focus only on sequence or structure, HEENN combines local and global features from both PAN and PPI networks. It uses a fully end-to-end architecture with attention and autoencoding mechanisms. This design greatly improves prediction accuracy.

## III. Preliminaries

### A. Graph Attention Network

The Graph Attention Network (GAT) is an extension of Graph Neural Networks (GNNs) that incorporates an attention mechanism to dynamically adjust the importance of neighboring nodes. Traditional GNNs treat all neighboring nodes equally without taking into account their varying significance. By adding an attention mechanism, GATs can learn the relative importance of each neighbor's features and assign different attention weights accordingly. This dynamic weighting enables the model to more accurately capture local structures and contextual information, thus enhancing its flexibility and performance [23], [24], [25].

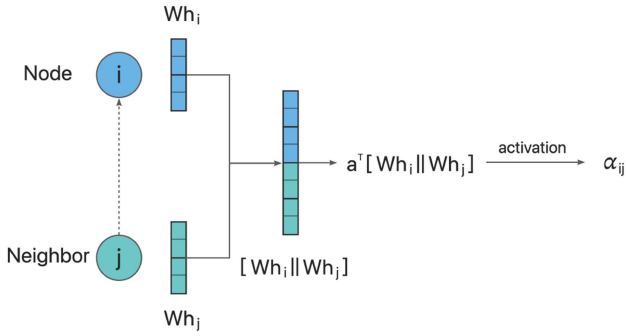


Fig. 1. Model of attention.(a) is a vector of attention parameters.

Fig. 1 depicts the computation flow of a single-head attention mechanism within the Graph Attention Network (GAT). The process begins with each node $h_i$ and its neighboring nodes $h_j$, where the goal is to compute an updated representation for $h_i$. This is done by aggregating information from its neighbors using learned attention weights. The calculation process described in the figure is as follows:

- *Linear Transformation:* Each node feature $h_i$ is linearly transformed into a new feature space via a shared weight matrix $W$, resulting in $Wh_i$. This operation is applied to all nodes in the graph [5], [26].
- *Concatenation and Attention Score Computation:* For each edge $(i, j)$, the transformed features of node i and its neighbor j are concatenated to get $[Wh_i|Wh_j]$. Then, this vector is passed through a shared attention vector $\mathbf{a}$, followed by a LeakyReLU activation. This process produces the unnormalized attention coefficient [27], [28].
- *Normalization via Softmax:* The attention coefficients $a_{ij}$ are normalized across all neighbors of $h_i$ using a softmax function to obtain the attention weights [29], [30].

These steps form the core of the GAT, enabling the model to focus dynamically on the most important neighbors and effectively capture critical relational information in the graph. By integrating attention mechanisms into the GNN framework, GAT enables more refined and context-sensitive feature extraction. This enhancement improves the model's ability to predict protein-protein interactions and perform other graph-based learning tasks [31].

### B. AutoEncoder

The AutoEncoder is a powerful neural network model that excels in learning efficient global feature representations by encoding and decoding information. In our model, we specifically utilize only the encoding part of the AutoEncoder, focusing on global feature extraction from the interaction data. The main strength of the AutoEncoder is its ability to automatically extract important features from raw, unlabelled data. It reduces dimensionality while preserving complex patterns and relationships. This makes it especially effective in cases where labeled data is limited, as it can learn useful representations without the need for supervision [32], [33], [34].

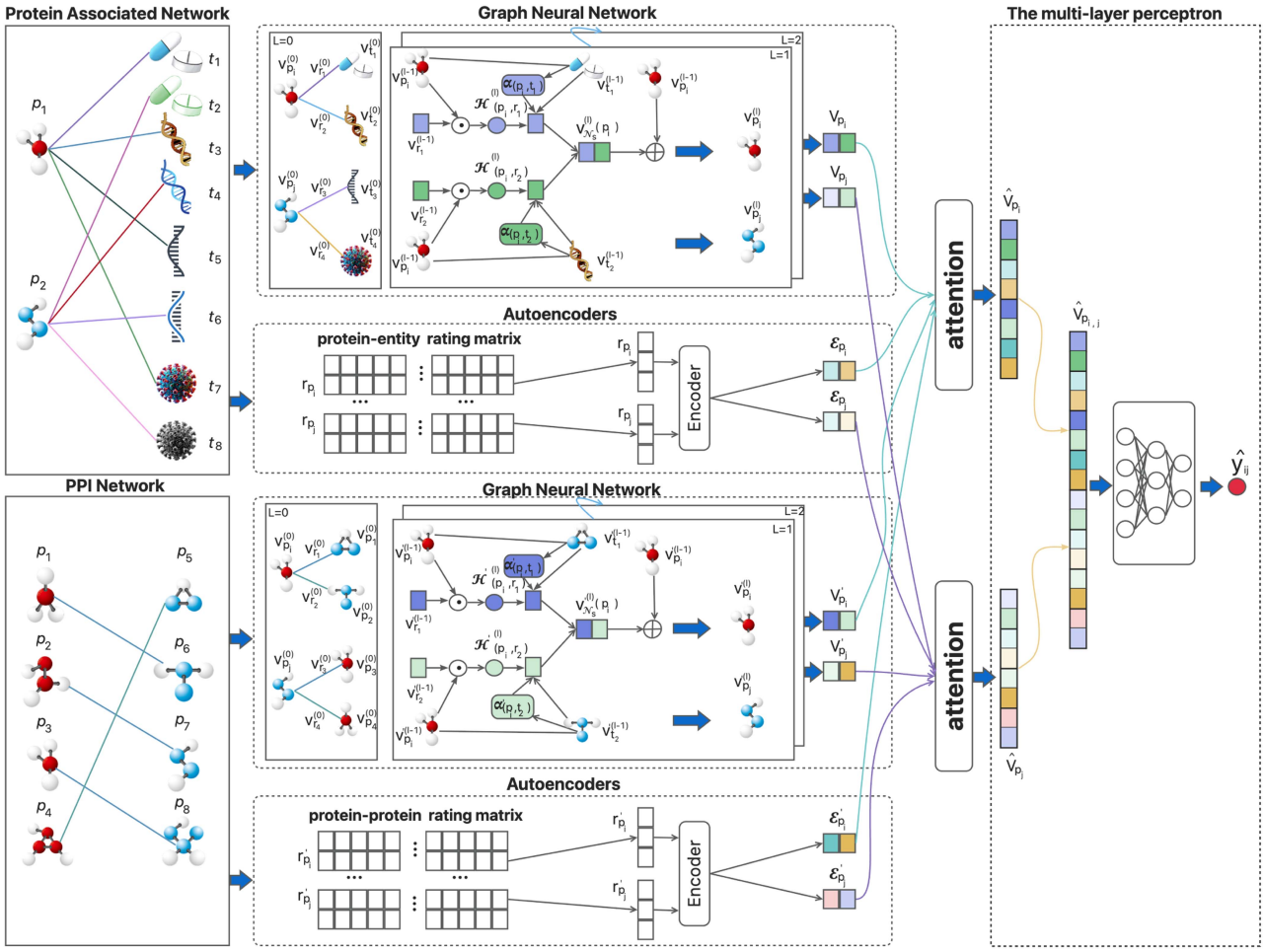


Fig. 2. Architecture of HEENN.

In the context of protein interactions, the AutoEncoder is used to capture global relationships between proteins and various biological entities such as drugs, diseases, and ribonucleic acids. By learning these interactions, the AutoEncoder enables the model to understand the broader context of protein-protein interactions (PPIs) and their associations with other biological processes.

The implementation of the AutoEncoder involves two key steps:

- *Constructing the Rating Matrix:* This step involves creating a rating matrix to capture the interactions between proteins and various biological entities. The matrix is structured to effectively represent these relationships, encoding the interactions as values in the matrix. In this way, the matrix becomes a comprehensive representation of how proteins are associated with other biological components, such as diseases or drugs. It provides a complete view of these complex interactions. This matrix forms the foundation for the learning process, capturing both direct and indirect relationships [35], [36].
- *Encoding the Rating Matrix:* After the rating matrix is constructed, the AutoEncoder's encoder network compresses it into a lower-dimensional latent space. This encoding process allows the model to capture the most relevant global features while reducing the complexity of the original data. By mapping the data into a more compact representation, the AutoEncoder retains the key relationships and patterns that define interactions between proteins and biological entities. These features are then available for downstream tasks like prediction. This compression reduces dimensionality and also helps identify and preserve the most important aspects of the data needed for accurate prediction [37], [38].

These two steps enable the AutoEncoder to concentrate on extracting global features. This ability to understand global information is crucial for improving the accuracy and depth of protein-protein interaction (PPI) predictions and allows the model to handle complex, large-scale biological data more effectively. Ultimately, the AutoEncoder can learn meaningful features from raw data without needing labeled examples. This ability significantly improves the model's performance in predicting protein interactions and related biological processes.

## IV. The Proposed HEENN

As shown in Fig. 2, the HEENN model consists of five key components: two Graph Attention Networks (GATs), two AutoEncoders, and a multi-layer perceptron (MLP), each of which

plays a crucial role in predicting protein-protein interactions (PPIs).

First, PAN is constructed by integrating relationships between proteins and biological entities like drugs, diseases, RNA, and protein structures. The first GAT extracts topological and semantic features from PAN, enhanced by an attention mechanism to focus on key relationships. A second GAT is then built on the PPI Network to capture protein-protein interaction features. It uses its own attention mechanism to adjust the importance of each interaction. This helps the model better identify critical relationships between proteins.

Next, to capture the global features of PAN and PPI, two AutoEncoders are employed. The first AutoEncoder processes the PAN dataset to extract global features, while the second AutoEncoder focuses on the PPI dataset, capturing the global interaction patterns between proteins. Both AutoEncoders help the model learn richer, more comprehensive representations of the data, complementing the GATs' local feature extraction.

Finally, a multi-layer perceptron (MLP) integrates the features from the two GATs and AutoEncoders. The MLP incorporates its own attention mechanism to selectively combine local and global features, ensuring that the most critical features contribute to the final prediction. This attention-based feature fusion enables the model to predict protein-protein interactions with higher accuracy.

In the following subsections, we will provide a detailed explanation of how the combination of GATs, AutoEncoders, and MLP leads to superior performance in PPI prediction.

### A. The Graph Attention Network

The first GAT (Graph Attention Network) generates protein feature representations. It does this by extracting topological and semantic features from the complex relationships within the Protein Associated Network (PAN). These relationships include interactions between proteins and various biological entities, such as drugs, diseases, ribonucleic acids, and protein structures. The second GAT component complements the first by incorporating PPI data to capture additional local topological features and interactions without the PAN. The combination of these two GAT components enables the model to accurately predict protein-protein interactions by focusing on the most relevant local features. The following section provides a brief overview of the datasets used for the GAT components.

*Protein Associated Network:* To build the PAN, we integrated four datasets capturing interactions between proteins and biological entities: miRNAs, diseases, drugs, and LncRNAs. Each interaction is represented as a triplet in the format *(protein, relation, entity)*. For example, an interaction between protein *9606.ensp00000284981* and LncRNA *nonhsat017939.2* is written as *(9606.ensp00000284981, Protein-LncRNA, nonhsat017939.2)*. The triplets cover four relation types: Protein-miRNA, Protein-Disease, Protein-Drug, and Protein-LncRNA. This structure enables PAN to capture both topological and semantic features, modeling complex relationships between proteins and various biological entities.

*PPI Network:* The PPI Network shares the same triplet format as PAN, representing protein-protein interactions as *(protein, relation, protein)*. Relation values are either 0 (no interaction) or 1 (interaction). For instance, an interaction between protein *9606.ensp00000267163* and itself is written as *(9606.ensp00000267163, 1, 9606.ensp00000267163)*. The PPI Network complements the PAN by incorporating protein interaction data, thereby enhancing information extraction and improving model performance.

*Learning process of GAT:* First, we utilize the Xavier initialization method to initialize the embedding representations of proteins. The initial embedding representation of the protein knowledge graph $\mathcal{G}$ is as follows:

$$V_{\mathcal{G}} = \left[ \underbrace{v_{p_1}^{(0)}, \ldots, v_{N_p}^{(0)}}_{protein\ embedding}, \underbrace{v_{r1}^{(0)}, \ldots, v_{N_r}^{(0)}}_{relation\ embedding}, \underbrace{v_{t1}^{(0)}, \cdots v_{N_t}^{(0)}}_{entity\ embedding} \right] \tag{1}$$

In formula (1), $N_p$, $N_r$, and $N_{\mathrm{t}}$ represent the number of proteins, relations, and entities, respectively. $v_p^{(0)} \in \mathbb{R}^d$, $v_r^{(0)} \in \mathbb{R}^d$, and $v_t^{(0)} \in \mathbb{R}^d$ represent the initialized embedding representations of proteins, relations, and entities, respectively. Here, $d$ represents the dimension of the embedding representation.

For a protein node $p_i$ in the knowledge graph $\mathcal{G}$, it is essential to randomly sample its neighboring nodes. The set of neighboring nodes obtained through this sampling is denoted as $\mathcal{N}_s(p_i)$. This random sampling strategy helps prevent overfitting by reducing the model's dependence on fixed neighborhood structures. It introduces stochasticity during training, encouraging the model to learn more general and robust representations. Moreover, random sampling decreases the likelihood of repeatedly selecting noisy or outlier nodes, thereby minimizing their negative impact on the learned features and enhancing overall robustness. Since each neighbor is connected to $p_i$ through a specific semantic relation $r$ (protein–protein, protein–drug, protein–miRNA, protein–disease, or protein–lncRNA interactions), this process also implicitly determines which types of relations are included in the current computational graph.

Consider a triple $(p_i, r_{in}, t_n)$, where $p_i$ represents the *i-th* protein, $r_{in}$ denotes the relation between the *i-th* protein and the *n-th* entity, and $t_n$ denotes the *n-th* entity. To aggregate the information from the edges surrounding the protein node, we combine the embedding representations of $p_i$ and $r_{in}$ using the following formula:

$$\mathcal{H}_{(p_i, r_{in})}^{(l)} = sum\left[\left(v_{p_i}^{(l-1)} \odot v_{r_{in}}^{(l-1)}\right) W_1 + b_1\right] \tag{2}$$

In the equation, $v_{r_{in}}^{(l-1)}$ represents the embedding representation of the relation between the protein $p_i$ and the entity $t_n$ in the $(l-1)^{th}$ layer of the GAT. $v_{p_i}^{(l-1)}$ represents the embedding representation of the protein $p_i$ in the $(l-1)^{th}$ layer of the GAT. $W_1 \in \mathbb{R}^{(d)\times d}$ represents the trainable weight matrix, $b_1$ represents the bias vector, and $p$ represents the number of layers in the fully connected layer. The symbol $\odot$ represents the Hadamard product.

In our model, we employ an attention mechanism to extract more relevant information from neighboring nodes. By weighting the embedding vectors of each neighbor, we assign an attention score to $t_n$, which indicates its importance to the protein $p_i$. Specifically, the attention value for the entity $t_n$ relative to the protein $p_i$ is calculated using the following formula:

$$e_{(p_i,t_n)}^{(l-1)} = LeakyReLU\left(W_3\left(W_2 v_{p_i}^{(l-1)} \oplus W_2 v_{t_n}^{(l-1)}\right)\right) \quad (3)$$

In this equation, $W_2 \in \mathbb{R}^{(d)\times 2\,d}$ and $W_3 \in \mathbb{R}^{(4\,d)}$ represents the trainable weight matrix. LeakyReLU is a activation function. $v_{t_n}^{(l-1)}$ represents the embedding representation of entity $t_n$. $\oplus$ represents the concatenation operation. Then, the attention values for all neighbors of each node are normalized, and the resulting normalized values represent the attention scores associated with each respective neighbor. The precise formula is given below:

$$\alpha_{(p_i,t_n)}^{(l)} = \text{softmax}\left(e_{(p_i,t_n)}^{(l-1)}\right) = \frac{\exp\left(e_{(p_i,t_n)}^{(l-1)}\right)}{\sum_{t_k\in\mathcal{N}_s(p_i)}\exp\left(e_{(p_i,t_k)}^{(l-1)}\right)} \quad (4)$$

In this equation, the symbol $exp$ refers to the exponential function.

Based on the aggregated edge information, the protein node and its neighboring entity nodes are integrated to generate the neighborhood embedding $v_{\mathcal{N}_s(p_i)}^{(l)}$ for protein $p_i$, as defined by the following formula:

$$v_{\mathcal{N}_s(p_i)}^{(l)} = \sum_{t_n\in\mathcal{N}_s(p_i)} \mathcal{H}_{(p_i,r_{in})}^{(l)} \alpha_{(p_i,t_n)}^{(l)} v_{t_n}^{(l-1)} \quad (5)$$

Finally, we aggregate the embedding representation $v_{p_i}^{(l-1)}$ and the neighborhood embedding representation $v_{\mathcal{N}_s(p_i)}^{(l)}$ of protein $p_i$ using the following aggregation function:

$$V_{p_i} = v_{p_i}^{(l)} = \sigma\left(\left(v_{p_i}^{(l-1)} \oplus v_{\mathcal{N}_s(p_i)}^{(l)}\right) W_4 + b_2\right) \quad (6)$$

In this equation, $W_4 \in \mathbb{R}^{(2\,d)\times d}$ is the trainable weight matrix, and $b_2$ represents the bias vector. The activation function $\sigma$ is applied. Using the same method, the embedding representation $V_{p_j}$ for the protein $p_j$ can be calculated.

Furthermore, both the PAN and PPI Network have a similar format. In the second part of GAT, we extract the embedding representation of the protein from the PPI Network using the same calculation method as in the first part of GAT. Within the second part of GAT, we obtain the embedding representations $V'_{p_i}$ and $V'_{p_j}$ for the proteins $p_i$ and $p_j$, respectively. These representations supplement the embedding representations $V_{p_i}$ and $V_{p_j}$ generated by the first part of GAT, effectively filling in the missing information within the PAN. Finally, the model's ability to represent the local features of proteins is enhanced by combining the protein embedding representations produced by the two GATs.

### B. The AutoEncoder

The first AutoEncoder processes the Protein Associated Network (PAN) dataset. It learns global protein feature representations by capturing complex relationships and patterns within the network. This AutoEncoder extracts high-level features that reflect the wider context of protein interactions, including associations with drugs, diseases, RNA, and protein structures.The second AutoEncoder focuses on the PPI dataset. It captures global interaction patterns between proteins and refines the feature representations. These refined features complement those learned from the PAN dataset. By combining these two AutoEncoders, the model can capture both the global relationships in biological data and the detailed interactions in the protein-protein interaction network. The dataset used by the AutoEncoders is the same as the one used for the GAT components, which was described earlier.

In protein-protein interaction (PPI) prediction, we use the Protein Associated Network (PAN) dataset to obtain information about protein-entity interactions. Additionally, the PPI dataset is employed to gather information about protein-protein interactions. These datasets are leveraged to construct interaction matrices that provide insights into both protein-entity and protein-protein relationships. The PAN dataset provides details about interactions between proteins and various biological entities, such as drugs, diseases, and ribonucleic acids. Using this information, we construct a partially observed interaction matrix $R \in \mathbb{R}^{m\times n}$, where $m$ represents the number of proteins and $n$ represents the number of biological entities. Each protein $p_i \in P = \{1,\ldots,m\}$ is represented by a partially observed vector $r_{p_i} = \{R_{(p_i,1)},\ldots,R_{(p_i,n)}\} \in \mathbb{R}^n$, where $R_{(p_i,j)} = 1$ indicates an interaction between protein $p_i$ and entity $t_j$, and $R_{(p_i,j)} = 0$ represents no interaction. Similarly, a protein interaction matrix based on the PPI network can also be constructed.

The AutoEncoder is designed to learn the missing protein-entity interactions. By encoding the interaction matrix obtained from the PAN dataset, the AutoEncoder captures and improves the representation of global protein features. Through the following formula, we can encode the protein vector representations $r_{p_i}$ from the interaction matrix into feature vectors that represent the proteins' characteristics:

$$\mathcal{E}_{p_i} = \sigma\left(r_{p_i} W_5^{(q)} + b_3^{(q)}\right) \quad (7)$$

In the equation, $W_5^{(q)}$ represents the trainable weight matrix, and $b_3^{(q)}$ represents the bias vector. The variable $q$ corresponds to the number of layers in the multi-layer perceptron and the activation function $\sigma$ represents the sigmoid function.

Moreover, the PAN and PPI Network exhibit a comparable structure. In the second phase of the AutoEncoder, the embedding representation of the protein is extracted from the PPI Network using the same method employed in the initial phase of the AutoEncoder. During this phase, the model obtains the embedding representations $\mathcal{E}p_i'$ and $\mathcal{E}p_j'$ for proteins $p_i$ and $p_j$. These new embeddings are used to supplement the earlier representations $\mathcal{E}p_i$ and $\mathcal{E}p_j$, which were generated in the first phase of the AutoEncoder. This supplementation helps address

the missing information in the PAN dataset. By combining the embeddings from both AutoEncoders, the model improves its ability to capture the global features of proteins.

### C. The Multi-Layer Perceptron

The multi-layer perceptron fuses features from two GATs and two AutoEncoders. It uses end-to-end learning for PPI prediction, based on integrated local and global features.

In the multi-layer perceptron (MLP) of the HEENN model, an attention mechanism is incorporated to adaptively weight the contributions of the different feature representations extracted from the two GATs and two AutoEncoders. Since the GAT trained on the PPI dataset performs the best, we treat it as the most informative. Therefore, we use the features it extracts, denoted as $V'_{p_i}$, as the reference. These features serve as the standard for computing the attention scores of the other three feature sets: $V_{p_i}$, $\mathcal{E}p_i$, and $\mathcal{E}p_i{}'$. Specifically, the attention mechanism measures the contribution of features from the other three components. These include the second GAT and the two AutoEncoders. It does this by comparing each of them to the reference features from the PPI-based GAT. This allows the model to focus more on the most relevant features from the other components, enhancing the overall prediction accuracy and ensuring the effective integration of local and global features. The relevant formulas for calculating these attention scores are introduced below:

$$\hat{e}_{V_{p_i}} = (W_6 V_{p_i} + b_4) \odot V'_{p_i} \tag{8}$$

In the equation, $W_6$ represents the trainable weight matrix, and $b_4$ represents the bias vector. $V_{p_i}$ represents the embedding generated by the first part of the GAT for protein $p_i$, $V'_{p_i}$ represents the embedding generated by the second part of the GAT for protein $p_i$. The symbol $\odot$ represents the Hadamard product. That indicate the importance of feature $V_{p_i}$ to feature $V'_{p_i}$ feature.

$$\hat{\alpha}_{V_{p_i}} = \frac{\exp\left(\hat{e}_{V_{p_i}}\right)}{\exp\left(\hat{e}_{V_{p_i}}\right) + \exp\left(\hat{e}_{\mathcal{E}_{p_i}}\right) + \exp\left(\hat{e}_{\mathcal{E}'_{p_i}}\right)} \tag{9}$$

In the equation, $\mathcal{E}_{p_i}$ represents the embedding generated by the first part of the AutoEncoder for protein $p_i$, $V'_{\mathcal{E}_i}$ represents the embedding generated by the second part of the AutoEncoder for protein $p_i$. The symbol $exp$ refers to the exponential function. $\hat{\alpha}_{V_{p_i}}$ represents the attention scores for the features derived from the first GAT.

Then, the four feature representations generated by the two GAT parts and the two AutoEncoder parts for each protein are fused together through the attention mechanism, resulting in the final embedding representation for protein $p_i$. The specific formula is as follows:

$$\hat{V}_{p_i} = \hat{\alpha}_{V_{p_i}} V_{p_i} \oplus V'_{p_i} \oplus \hat{\alpha}_{\mathcal{E}_{p_i}} \mathcal{E}_{p_i} \oplus \hat{\alpha}_{\mathcal{E}'_{p_i}} \mathcal{E}'_{p_i} \tag{10}$$

In the equation, $\hat{V}_{p_i}$ represents the final embedding representation for protein $p_i$. The symbol $\oplus$ represents the concatenation operation.

In a similar manner, the final embedding representation $\hat{V}_{p_j}$ for protein $p_j$ can be obtained.

Finally, a multi-layer perceptron is utilized to predict the interaction between the two proteins, leveraging their respective final feature representations. The corresponding formula is as follows:

$$\hat{\mathcal{Y}}_{ij} = \sigma\left((\hat{V}_{p_i} \oplus \hat{V}_{p_j}) W_7^{(p)} + b_5^{(p)}\right) \tag{11}$$

In the equation, $W_7^{(p)}$ represents the trainable weight matrix, and $b_5^{(p)}$ represents the bias vector. The symbol $\oplus$ denotes the concatenation operation and the variable $p$ corresponds to the number of layers in the multi-layer perceptron. The activation function $\sigma$ represents the sigmoid function and $\hat{\mathcal{Y}}_{ij}$ represents the final predicted score.

During the model optimization process, the cross-entropy loss function is used. The expression for the cross-entropy loss is as follows:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{12}$$

In the formula, $N$ is the total number of samples. $y_i$ is the true label of the $i$-th sample, taking a value of 0 or 1. $\hat{y}_i$ is the probability that the $i$-th sample is predicted to belong to the positive class.

Furthermore, batch normalization layers are incorporated following each fully connected layer to expedite the model's convergence. To reduce overfitting and improve model performance, dropout layers and $\ell_2$ regularization are also applied.

### D. Algorithm Design and Complexity Analysis

The working principle of the model is thoroughly outlined in the preceding section, and the corresponding pseudo-code is summarized in Algorithm 1.

The time complexity of the model is expressed as $O(E \times \frac{D}{B} \times T)$, where $E$ represents the number of epochs, $D$ represents the size of the dataset, $B$ represents the batch size, and $T$ represents the time complexity of each iteration. Assuming that the dimension of the first GAT is denoted as $G_1$, the dimension of the second GAT is denoted as $G_2$, the dimension of the first AutoEncoder is denoted as $G_3$, the dimension of the second AutoEncoder is denoted as $G_4$, then the time complexity of one iteration is estimated as $O(B \times (G_1 + G_2 + G_3 + G_4)^2)$. Consequently, the final time complexity of the model is approximated as $O(E \times D \times (G_1 + G_2 + G_3 + G_4)^2)$.

## V. Experiments

In the following experiments, we seek to address the research questions outlined below (RQs):

- RQ. 1. Does the proposed HEENN model achieve superior performance compared to existing state-of-the-art methods in predicting PPIs?
- RQ. 2. What types of structures can the HEENN model effectively identify within the protein interaction graph?
- RQ. 3. How do the different substructures in the HEENN model and the dataset influence the overall prediction accuracy?

**Algorithm 1:** The HEENN Algorithm.

**Input:** PAN $\mathcal{G}$(N, E), PPI Network $\mathcal{G}'(N',\ E')$, PAN Interaction Matrix $R \in \mathbb{R}^{m\times n}$, PPI Interaction Matrix $R' \in \mathbb{R}^{m\times n}$

**Output:** Prediction score $\hat{\mathcal{Y}}_{ij}$ .

initialization;

**for each** $n_i\ n_j \in N\ ,n'_i\ n'_j \in N'\ ,r_i\ r_j \in R_i\ ,r'_i\ r'_j \in R'_i$ **do**

  **for each** $n_i,n_j,n'_i,n'_j r_i,r_j,r'_i,r'_j$ **do**

    $\mathcal{N}_s(n) \leftarrow NeighborhoodSampling(n)$;

    $\mathcal{H}_{(n)} \leftarrow sum[(v_p \odot v_r)W_1 + b_1]$;

    $e_{(n)} \leftarrow LeakyReLU(W_3(W_2 v_p \oplus W_2 v_t))$;

    $\alpha_{(n)} \leftarrow softmax(e_{(n)})$;

    $v_{\mathcal{N}_s(v)} \leftarrow \sum_{t\in\mathcal{N}_s(v)} \mathcal{H}_{(n)}\alpha_{(n)} v_t$;

    $V_n \leftarrow v_n = \sigma((v_n \oplus v_{\mathcal{N}_s(v)})W_4 + b_2)$;

    $\mathcal{E}_n \leftarrow \sigma(r_i W_5 + b_3)$;

  **for each** $V_{n_i},\mathcal{E}_{n_i},\mathcal{E}'_{n_j}$ **do**

    $\hat{e}_{V_n} \leftarrow (W_6 V_n + b_4) \odot V_{n'_i}$;

  $\hat{\alpha}_{V_{n_i}} \leftarrow \frac{exp(\hat{e}_{V_{n_i}})}{exp(\hat{e}_{V_{n_i}})+exp(\hat{e}_{\mathcal{E}_{r_i}})+exp(\hat{e}_{\mathcal{E}_{r'_i}})}$;

  $\hat{\alpha}_{\mathcal{E}_{r_i}} \leftarrow \frac{exp(\hat{e}_{\mathcal{E}_{r_i}})}{exp(\hat{e}_{V_{n_i}})+exp(\hat{e}_{\mathcal{E}_{r_i}})+exp(\hat{e}_{\mathcal{E}_{r'_i}})}$;

  $\hat{\alpha}_{\mathcal{E}_{r'_i}} \leftarrow \frac{exp(\hat{e}_{\mathcal{E}_{r'_i}})}{exp(\hat{e}_{V_{n_i}})+exp(\hat{e}_{\mathcal{E}_{r_i}})+exp(\hat{e}_{\mathcal{E}_{r'_i}})}$;

  $\hat{V}_{n_i} \leftarrow \hat{\alpha}_{V_{n_i}} V_{n_i} \oplus V_{n'_i} \oplus \hat{\alpha}_{\mathcal{E}_{r_i}} \mathcal{E}_{r_i} \oplus \hat{\alpha}_{\mathcal{E}_{r'_i}} \mathcal{E}_{r'_i}$;

  $\hat{V}_{n_j} \leftarrow \hat{\alpha}_{V_{n_j}} V_{n_j} \oplus V_{n'_j} \oplus \hat{\alpha}_{\mathcal{E}_{r_j}} \mathcal{E}_{r_j} \oplus \hat{\alpha}_{\mathcal{E}_{r'_j}} \mathcal{E}_{r'_j}$;

  $\hat{\mathcal{Y}}_{ij} \leftarrow \sigma((\hat{V}_{n_i} \oplus \hat{V}_{n_j})W_7 + b_5)$;

  $\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$;

- RQ. 4. What is the impact of various hyperparameters on the performance of the HEENN model?

### A. General Settings

*Datasets:* We employ two datasets in our experiments, referred to as Dataset1 and Dataset2, each comprising a Protein-Associated Network (PAN) and a Protein-Protein Interaction (PPI) network.

In Dataset1, both PAN and PPI are derived from MTV-PPI [39], where the original data undergo preprocessing through deletions and reconstructions to form a refined version. In contrast, Dataset2 is constructed from publicly available biological databases: protein-protein interactions are sourced from the SPRING database [40], while protein-entity associations in PAN are collected from DrugBank [41], LncTarD [42], miRTarBase [43], and CTD [44].

For both datasets, the PAN comprises four types of protein-related associations: Protein–Drug, Protein–LncRNA, Protein–miRNA, and Protein–Disease. The PAN in Dataset1 includes 41,828 triplets involving 1,649 proteins and 1,944 associated biological entities, while Dataset2's PAN contains 56,759 triplets, 4,283 proteins, and 13,596 entities. The PPI network in Dataset1 consists of 19,237 triplets and 1,466 proteins, whereas Dataset2 includes 52,034 triplets with 4,283 proteins. The PPI datasets only contain positive interactions, and during training, an equal number of negative triplets are generated by randomly pairing non-interacting proteins and labeling them with a relation type of 0. For both datasets, all PAN triplets are used for training, while the PPI data is split into 80% for training and 20% for testing.

TABLE I
STATISTICS OF THE STUDIED DATASETS FROM DATASET1 AND DATASET2

| Dataset | Subpart | Protein | Relation | Entity | Triplet |
|---|---|---|---|---|---|
| Dataset1 | Protein–Drug | 613 | 1 | 969 | 11107 |
| | Protein–LncRNA | 164 | 1 | 12 | 690 |
| | Protein–miRNA | 489 | 1 | 271 | 4944 |
| | Protein–Disease | 832 | 1 | 692 | 25087 |
| | **All (PAN)** | **1649** | **4** | **1944** | **41828** |
| | **Protein–Protein (PPI)** | **1466** | **1** | **–** | **19237** |
| Dataset2 | Protein–Drug | 2226 | 1 | 7079 | 19057 |
| | Protein–LncRNA | 1304 | 1 | 1108 | 4826 |
| | Protein–miRNA | 1965 | 1 | 952 | 7303 |
| | Protein–Disease | 3899 | 1 | 4457 | 25573 |
| | **All (PAN)** | **4283** | **4** | **13596** | **56759** |
| | **Protein–Protein (PPI)** | **4283** | **1** | **–** | **52034** |

**PAN** = Protein Association Network, which combines four relation types: Protein–Drug, Protein–LncRNA, Protein–miRNA, and Protein–Disease. **PPI** = Protein–Protein Interaction. The bold values highlight the aggregated statistics for two special cases: 1. All (PAN) — the combined Protein Association Network, which merges all four relation types (Protein–Drug, Protein–LncRNA, Protein–miRNA, and Protein–Disease). The bold numbers here represent: • Protein: total unique proteins across all four relation types. • Relation: total number of relation types (always 4 here). • Entity: total unique entities across the four types. • Triplet: total number of triplets in the combined network. 2. Protein–Protein (PPI) — the subset containing only protein–protein interaction data. The bold numbers here show: • Protein: total proteins involved in PPI. • Relation: total relation types (1 here, because it's only protein–protein). • Triplet: total number of PPI triplets. (Entity is not shown because it overlaps with Protein in PPI.) In short, bold values indicate summed totals or key subset statistics rather than per-subpart counts.

Summary statistics for both Dataset1 and Dataset2 are provided in Table I, where bold values indicate aggregated statistics for All (PAN) and Protein–Protein (PPI) subsets.

*Evaluation Metrics:* PPI prediction is a binary classification problem, and the model's performance is evaluated using several metrics: Accuracy (Acc), Recall (Rec), Precision (Pre), Matthews Correlation Coefficient (MCC), Area Under the Curve (AUC), and Area Under the Precision-Recall Curve (AUPR).

*Baselines:* We compare HEENN with eleven state-of-the-art related models: WSRC_GE [45], LR_PPI [46], DPPI [47], PIPR [15], LPPI [48], MDNN [49], MTV-PPI [39], TAGPPI [50], HNSPPI [51], EResCNN [52], and KGF-GNN [13]. Please refer to the Appendix, available online, to see more details.

*Implementation Detail:* A comprehensive evaluation was performed using five-fold cross-validation across all models. The final results were derived by averaging the outcomes of the five iterations. Furthermore, the standard deviation was computed to assess the consistency and stability of the models. The code is written in Python3.9 and pytorch, pandas, numpy, sklearn, etc are also used. The experiments were performed on a computer with i5-12400F CPU and GTX4060 GPU.

### B. Performance Comparison (RQ. 1)

The proposed model, HEENN, was compared with state-of-the-art models, with detailed comparison results presented in Table II. To evaluate the model's performance across multiple dimensions, we utilized a range of evaluation metrics, including

TABLE II
MODEL COMPARISON RESULTS ON DATASET1 AND DATASET2

| Dataset | Model | Acc | Rec | Pre | AUC | AUPR | MCC | Win/Loss | *p*-value | F-rank |
|---|---|---|---|---|---|---|---|---|---|---|
| Dataset1 | LR_PPI | 0.7717±0.0066 | 0.7551±0.0090 | 0.7329±0.0092 | 0.8482±0.0060 | 0.8411±0.0058 | 0.6027±0.0089 | 6/0 | 0.0156 | 11.00 |
| | DPPI | 0.8007±0.0087 | 0.7623±0.0099 | 0.7677±0.0090 | 0.8726±0.0076 | 0.8903±0.0078 | 0.6334±0.0097 | 6/0 | 0.0156 | 9.58 |
| | WSRC_GE | 0.8225±0.0105 | 0.7623±0.0097 | 0.7987±0.0123 | 0.9022±0.0089 | 0.8975±0.0086 | 0.6996±0.0118 | 6/0 | 0.0156 | 7.42 |
| | LPPI | 0.8062±0.0116 | **0.9275±0.0124** | 0.7232±0.0103 | 0.8424±0.0173 | 0.8022±0.0154 | 0.6779±0.01587 | 5/1 | 0.0313 | 8.83 |
| | PIPR | 0.7536±0.0090 | 0.7678±0.0100 | 0.7456±0.0098 | 0.8331±0.0094 | 0.8246±0.0096 | 0.6623±0.0097 | 6/0 | 0.0156 | 10.50 |
| | MTV-PPI | 0.8655±0.0050 | 0.8249±0.0085 | 0.8979±0.0088 | 0.9301±0.0050 | 0.9308±0.0045 | 0.6838±0.0079 | 6/0 | 0.0156 | 4.83 |
| | MDNN | 0.8337±0.0027 | 0.8607±0.0082 | 0.8167±0.0071 | 0.8998±0.0009 | 0.8823±0.0046 | 0.6595±0.0063 | 6/0 | 0.0156 | 7.83 |
| | HNSPPI | 0.9031±0.0033 | 0.9040±0.0050 | 0.8797±0.0063 | 0.9031±0.0034 | 0.9233±0.0030 | 0.8078±0.0065 | 6/0 | 0.0156 | 3.83 |
| | TAGPPI | 0.8739±0.0046 | 0.9015±0.0053 | 0.8544±0.0042 | 0.8738±0.0037 | 0.9026±0.0057 | 0.7488±0.0049 | 6/0 | 0.0156 | 5.33 |
| | EResCNN | 0.8480±0.0024 | 0.8613±0.0056 | 0.8390±0.0018 | 0.9223±0.0022 | 0.9218±0.0031 | 0.6963±0.0050 | 6/0 | 0.0156 | 5.50 |
| | KGF-GNN | 0.9065±0.0033 | 0.9087±0.0041 | 0.9048±0.0036 | 0.9548±0.0017 | 0.9460±0.0033 | 0.8207±0.0024 | 6/0 | 0.0156 | 2.17 |
| | **HEENN (ours)** | **0.9159±0.0034** | 0.9112±0.0120 | **0.9200±0.0065** | **0.9596±0.0020** | **0.9527±0.0047** | **0.8320±0.0067** | **65/1** | **-** | **1.17** |
| Dataset2 | LR_PPI | 0.8321±0.0045 | 0.8551±0.0078 | 0.8111±0.0082 | 0.9043±0.0078 | 0.8848±0.0048 | 0.7441±0.0098 | 6/0 | 0.0156 | 11.00 |
| | DPPI | 0.8648±0.0109 | 0.8789±0.0108 | 0.8488±0.0115 | 0.9247±0.0078 | 0.9089±0.0106 | 0.7789±0.0107 | 6/0 | 0.0156 | 9.33 |
| | WSRC_GE | 0.8798±0.0097 | 0.8988±0.0089 | 0.8589±0.0104 | 0.9332±0.0094 | 0.9189±0.0112 | 0.7989±0.0132 | 6/0 | 0.0156 | 8.33 |
| | LPPI | 0.8897±0.0123 | 0.9254±0.0121 | 0.8101±0.0112 | 0.8993±0.0161 | 0.8665±0.0164 | 0.8097±0.0149 | 6/0 | 0.0156 | 9.67 |
| | PIPR | 0.8256±0.0097 | 0.8708±0.0101 | 0.8287±0.0099 | 0.9031±0.0086 | 0.8841±0.0088 | 0.7598±0.0098 | 6/0 | 0.0156 | 11.00 |
| | MTV-PPI | 0.9132±0.0056 | 0.9301±0.0093 | 0.8977±0.0092 | 0.9534±0.0088 | 0.9575±0.0065 | 0.8502±0.0089 | 6/0 | 0.0156 | 4.67 |
| | MDNN | 0.9036±0.0042 | 0.9188±0.0076 | 0.8876±0.0087 | 0.9206±0.0076 | 0.9349±0.0078 | 0.8307±0.0083 | 6/0 | 0.0156 | 6.83 |
| | HNSPPI | 0.9403±0.0052 | 0.9382±0.0063 | 0.9289±0.0061 | 0.9501±0.0056 | 0.9511±0.0067 | 0.8842±0.0072 | 6/0 | 0.0156 | 3.83 |
| | TAGPPI | 0.9244±0.0067 | 0.9432±0.0062 | 0.9088±0.0074 | 0.9577±0.0054 | 0.9431±0.0051 | 0.8551±0.0061 | 6/0 | 0.0156 | 4.17 |
| | EResCNN | 0.9011±0.0035 | 0.9157±0.0048 | 0.8843±0.0034 | 0.9608±0.0057 | 0.9479±0.0043 | 0.8207±0.0062 | 6/0 | 0.0156 | 6.17 |
| | KGF-GNN | 0.9491±0.0045 | 0.9658±0.0032 | 0.9345±0.0035 | 0.9781±0.0016 | 0.9675±0.0037 | 0.9086±0.0031 | 6/0 | 0.0156 | 2.00 |
| | **HEENN (ours)** | **0.9696±0.0021** | **0.9773±0.0024** | **0.9553±0.0043** | **0.9859±0.0051** | **0.9868±0.0046** | **0.9357±0.0087** | **66/0** | **-** | **1.00** |

The bold values highlight the aggregated statistics for two special cases: 1. All (PAN) — the combined Protein Association Network, which merges all four relation types (Protein–Drug, Protein–LncRNA, Protein–miRNA, and Protein–Disease). The bold numbers here represent: • Protein: total unique proteins across all four relation types. • Relation: total number of relation types (always 4 here). • Entity: total unique entities across the four types. • Triplet: total number of triplets in the combined network. 2. Protein–Protein (PPI) — the subset containing only protein–protein interaction data. The bold numbers here show: • Protein: total proteins involved in PPI. • Relation: total relation types (1 here, because it's only protein–protein). • Triplet: total number of PPI triplets. (Entity is not shown because it overlaps with Protein in PPI.) In short, bold values indicate summed totals or key subset statistics rather than per-subpart counts.

accuracy (Acc), recall (Rec), precision (Pre), Matthews correlation coefficient (MCC), area under the receiver operating characteristic curve (AUC), and area under the precision-recall curve (AUPR). These metrics provide a comprehensive evaluation of the model's effectiveness. Additionally, to gain deeper insights into the experimental results, statistical analyses were conducted using methods such as the win/loss ratio, Wilcoxon signed-ranks test, and Friedman test.

The win/loss ratio quantifies the number of evaluation metrics in which the HEENN model demonstrates superior or inferior performance compared to baseline models. The Wilcoxon signed-ranks test, a nonparametric pairwise comparison method, evaluates whether HEENN achieves significantly higher prediction accuracy than each comparison model based on the significance level (p-value). The Friedman test is employed to compare the performance of multiple models across multiple datasets simultaneously using the F-rank value, where a lower F-rank value indicates higher prediction accuracy.

Additionally, the runtime of the proposed HEENN model was compared with that of state-of-the-art models using Dataset2. Each model was evaluated under identical experimental settings, and the average runtime required to complete the prediction task is shown in Fig. 3. It is evident that HEENN achieves lower runtime than most baseline methods, including the original KGF-GNN. Although some earlier models run faster due to simpler computations, they fall short in prediction accuracy. HEENN not only delivers superior performance but also maintains a relatively low runtime, making it a more efficient and scalable choice. A similar runtime comparison was also conducted on Dataset1, and the results were consistent. Details are provided in the Appendix, available online.

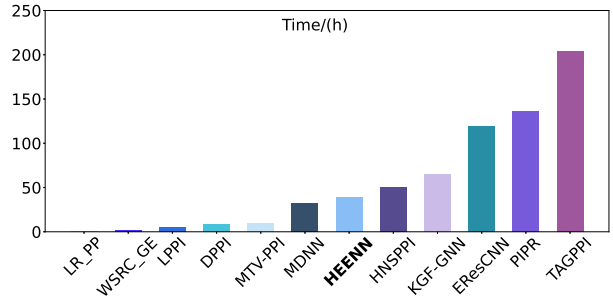


Fig. 3. Running times of different models.

As shown in the Table II, the HEENN model demonstrates superior performance across all evaluation metrics. A detailed analysis is provided below:

- Across the 132 evaluations conducted on both Dataset1 and Dataset2, the proposed HEENN model demonstrated exceptional and consistent performance with 131 wins and only 1 loss. On Dataset2, HEENN achieved a perfect 66/0 win/loss ratio and ranked first in all evaluation metrics, including Accuracy, Recall, Precision, AUC, AUPR, and MCC. On Dataset1, although the Recall of HEENN was

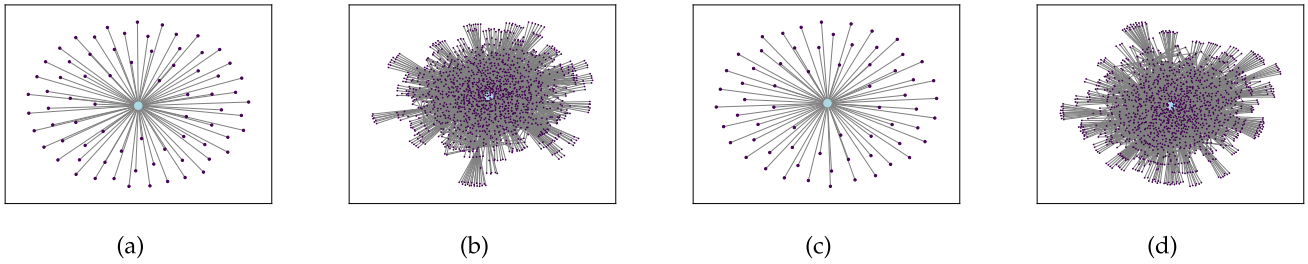


Fig. 4. The graph structures identified by the KGF-GNN model. (a) The first-order neighbor graph of the first node in the prediction sample. (b) The second-order neighbor graph of the first node in the prediction sample. (c) The first-order neighbor graph of the second node in the prediction sample. (d) The second-order neighbor graph of the second node in the prediction sample.

slightly lower than that of the LPPI model, it outperformed LPPI by at least 0.1 in the other five metrics. These results confirm the robustness and reliability of HEENN in protein-protein interaction prediction across different datasets. The superior performance of HEENN is attributed to its ability to integrate both local and global features through an end-to-end architecture enhanced by attention-based fusion. The attention mechanism enables HEENN to adaptively weigh and combine multi-source features, leading to more stable and accurate predictions across various metrics and tasks.

- Among all the models, the HEENN model achieves the highest F-rank, further confirming its superior predictive performance. This result establishes the HEENN model as the leading performer among the compared models.
- All the obtained p-values are below the 0.05 significance threshold, providing robust evidence that the predictive performance of the HEENN model surpasses that of all the compared models at a significance level of 0.05. This statistically significant difference further underscores the superior predictive capabilities of the HEENN model.
- Furthermore, it is noteworthy that the HEENN model exhibits a lower standard deviation compared to the majority of other models. This observation indicates that the HEENN model demonstrates exceptional stability in its predictions, reflecting consistent performance across different experiments or datasets.

These results validate the effectiveness of HEENN's architectural design. As an end-to-end neural network, HEENN overcomes common limitations of traditional feature engineering, such as error accumulation and information loss. It combines local topological and semantic features, extracted by Graph Attention Networks (GATs), with global contextual representations learned by AutoEncoders trained on Protein-Associated Networks (PANs) and PPI data. The attention-based fusion mechanism further ensures that these diverse features are integrated effectively, allowing the model to capture complex biological relationships and significantly improve the accuracy of protein-protein interaction prediction.

### C. Case Study (RQ. 2)

In this section, we analyze and compare the graph structures identified by the HEENN and KGF-GNN models. This helps to better understand how each model represents protein-protein interaction (PPI) networks. Since both models share a similar GNN-based backbone, KGF-GNN is used as a baseline for comparison. To support this comparison, we examine two types of local subgraphs: the first-order neighborhood graph, which includes the target protein and its directly connected proteins, and the second-order neighborhood graph, which further includes the neighbors of those direct connections. These graphs reflect how information is propagated in the network and reveal the structural patterns each model learns to recognize.

We visualize the first- and second-order neighborhoods for two proteins correctly predicted by the KGF-GNN model (Fig. 4), and for two proteins correctly predicted by HEENN but misclassified by KGF-GNN (Fig. 5). In each case, subfigures (a) and (b) show the first protein, and (c) and (d) the second. By comparing the two figures, we can see clear differences in graph complexity. The HEENN model identifies more intricate structures. Its graphs contain more nodes and more connections, especially in the second-order neighborhoods. In contrast, the KGF-GNN model captures simpler, sparser structures.

This difference comes from the models' architectures. HEENN combines local and global feature extraction using a GAT and an AutoEncoder. This helps it capture complex patterns beyond direct neighbors. KGF-GNN, however, uses only a single GNN, focusing on local features and missing global contexts. As a result, it struggles with complex structures and may misclassify some proteins. With its hybrid design, HEENN builds richer graph representations and delivers better predictive performance.

### D. Ablation Study (RQ. 3)

To investigate the impact of various components within the model and dataset on overall performance, an ablation study was conducted using Dataset2. Ablation experiments involve systematically removing specific elements to evaluate their contributions to the model's effectiveness. By isolating these components, we gain deeper insights into the mechanisms behind the model's performance.

This section presents two primary ablation experiments. The first evaluates the influence of the two GAT components and the two AutoEncoder components. The second examines the effects of different substructures within the PAN on predictive accuracy. All results are averaged over five-fold cross-validation to ensure reliability. A similar ablation study was also performed on Dataset1, showing consistent trends. Details can be found in the Appendix, available online.

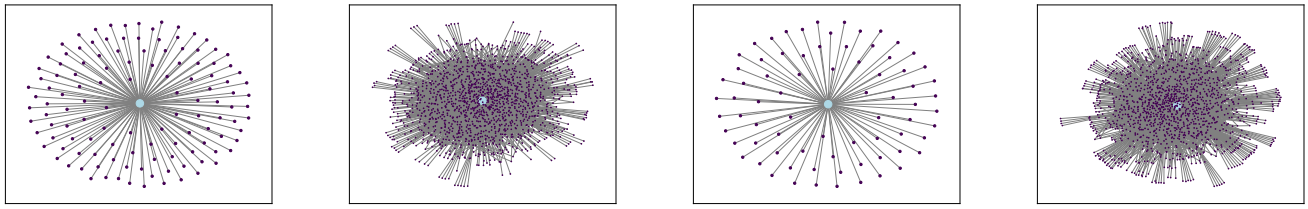


Fig. 5. The graph structures identified by the HEENN model. (a) The first-order neighbor graph of the first node in the prediction sample. (b) The second-order neighbor graph of the first node in the prediction sample. (c) The first-order neighbor graph of the second node in the prediction sample. (d) The second-order neighbor graph of the second node in the prediction sample.

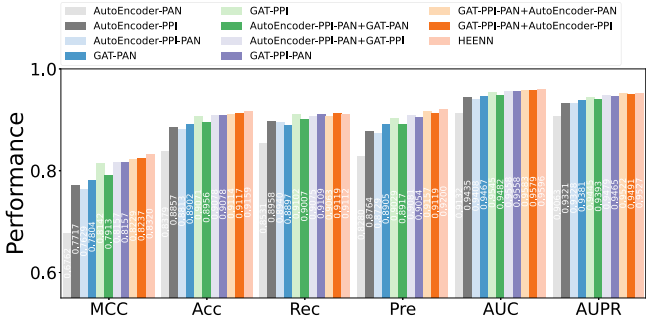


Fig. 6. Results of ablation experiments for the model.

*1) Ablation Experiments for the Model:* First, I provide explanations for some of the symbols used in Fig. 6. AutoEncoder-PAN and AutoEncoder-PPI use AutoEncoder modules to process PAN and PPI data individually, while GAT-PAN and GAT-PPI perform the same functions using Graph Attention Networks (GAT). AutoEncoder-PPI-PAN and GAT-PPI-PAN extend these approaches by using two modules (either AutoEncoders or GATs) to process both PPI and PAN data separately. AutoEncoder-PPI-PAN+GAT-PAN and AutoEncoder-PPI-PAN+GAT-PPI combine dual AutoEncoders with an additional GAT for either PAN or PPI. Conversely, GAT-PPI-PAN+AutoEncoder-PAN and GAT-PPI-PAN+AutoEncoder-PPI use dual GATs along with an extra AutoEncoder for PAN or PPI.

Based on these settings, the first ablation experiment was conducted to thoroughly analyze the individual and combined contributions of the AutoEncoder and GAT components in the HEENN model. The results are illustrated in Fig. 6. From the data, it is evident that the AutoEncoder components (AutoEncoder-PPI and AutoEncoder-PAN) contribute to performance enhancement but are less effective when used alone compared to the GAT components (GAT-PPI and GAT-PAN). Specifically, AutoEncoder-PPI consistently performs better than AutoEncoder-PAN across all metrics, which may be attributed to the richer PPI-related information encoded in the PPI network compared to the PAN network. However, the AutoEncoder components alone are unable to fully capture the complex interactions needed for accurate PPI prediction. This may be due to the limitations of the interaction matrix representation used in AutoEncoders, which lacks explicit graph structure and topological dependencies. Additionally, the unsupervised learning process makes it difficult for AutoEncoders to focus on interaction-relevant patterns without label guidance. As a result, their standalone performance is constrained. The GAT components exhibit significantly better performance than the AutoEncoder components, with GAT-PPI outperforming GAT-PAN in metrics such as MCC (by 0.02), AUC (by 0.01), and AUPR (by 0.01). This difference underscores the importance of leveraging the PPI network for extracting topological and semantic features directly related to protein-protein interactions. On the other hand, GAT-PAN's relatively lower performance reflects the limited contribution of protein-associated networks alone when not complemented by direct PPI information.

Notably, the combination of AutoEncoder and GAT components provides a noticeable performance boost. For instance, the GAT-PPI-PAN+AutoEncoder-PPI model achieves an average improvement of approximately 0.03 across most metrics compared to the standalone GAT-PPI model. This synergy indicates that the global feature extraction capabilities of AutoEncoders complement the local topological and semantic features captured by the GATs, leading to more robust protein representations. Furthermore, the GAT-PPI-PAN+AutoEncoder-PAN model shows a slight decline in performance compared to the GAT-PPI-PAN+AutoEncoder-PPI model, further affirming the richer predictive information in the PPI network. The fully integrated HEENN model, combining all components, achieves the best overall performance. It outperforms GAT-PPI-PAN+AutoEncoder-PPI across most metrics, with a 0.01 gain in MCC. The only exception is REC, where it performs slightly lower. This highlights the effectiveness of HEENN's design, where the attention-based feature fusion mechanism seamlessly integrates the strengths of both AutoEncoders and GATs. By capturing global protein interaction patterns and local graph features simultaneously, HEENN achieves a significant advantage over its individual components.

These results demonstrate that the combination of AutoEncoder and GAT components is critical for leveraging both global and local information in PPI networks. Moreover, the HEENN model's ability to adaptively fuse these features through its attention-enhanced architecture enables it to achieve state-of-the-art performance, making it a promising tool for bioinformatics applications.

*2) Ablation Experiments for Dataset PAN:* In the second ablation experiment, we specifically examined the entities within

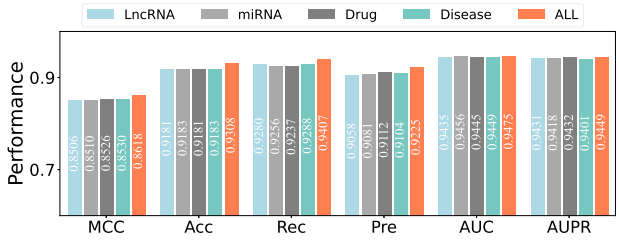


Fig. 7. Results of ablation experiments for PAN.

the PAN, focusing on the categories of Drug, LncRNA, miRNA, and Disease. Individual evaluations were conducted using the GAT-PAN model to assess the impact of each entity category on the model's performance. The experimental results are illustrated in Fig. 7. The data indicate that each entity category within the PAN dataset contributes comparably to the model's performance. Among them, Disease entities exhibit a slightly greater influence, suggesting a potentially stronger correlation between disease-related information and protein interactions. Moreover, integrating all four entity types to construct a complete PAN can improve the MCC, Acc, Rec, and Pre of the GAT-PAN model by about 0.01. This finding implies that incorporating all entity types provides richer topological and semantic features, thereby enhancing the model's overall predictive performance.

### *E. Parameter Sensitivity Analysis (RQ. 4)*

In this section, we analyze the influence of key hyperparameters on the model's performance using Dataset2. The hyperparameters under investigation include the coefficient of the $\ell_2$ regularization term, the embedding dimensions in the two GAT components, the size of neighborhood sampling, and the number of GAT layers. By adjusting these parameters and observing their impact, we gain a clearer understanding of how the model processes input data and which configurations lead to optimal performance.

The experimental results are presented in Fig. 8. Similar experiments were also conducted on Dataset1. Although some specific trends differ slightly, the general conclusions remain consistent. Detailed results and observations for Dataset1 are provided in the Appendix, available online.

*1) The Impact of the $\ell_2$ Regularization Coefficient on Model Performance:* From Fig. 8(c), we can see how the $\ell_2$ regularization coefficient affects model performance. When the coefficient is too large, the model becomes overly constrained. This limits its capacity to learn complex patterns in the data and leads to a drop in performance. On the other hand, when the coefficient is too small, the model tends to overfit. It performs well on the training set but generalizes poorly to unseen data. Notably, the performance shows consistent gains from 0.01 to 0.0001, with metrics such as ACC, AUC, and AUPR reaching their highest levels around 0.0001. However, beyond this point, improvements become marginal and the performance starts to plateau. To balance generalization and overfitting, we select 0.0001 as the optimal coefficient. This choice ensures robust generalization without sacrificing predictive power.

*2) The Impact of the Dimensionality of Embedding Representations on Model Performance:* Fig. 8(a) clearly shows that as the DDI embedding dimension increases, the model achieves steadily better results across all evaluation metrics. Notably, the Matthews correlation coefficient (MCC) shows the most significant improvement, increasing steadily as the embedding size expands. This suggests that a higher-dimensional representation enhances the model's ability to capture complex drug-drug interaction patterns, leading to more accurate predictions. However, the performance gains begin to plateau at larger dimensions, indicating diminishing returns with excessive embedding size.

In contrast, from Fig. 8(b), the effect of varying the Knowledge Graph (KG) embedding dimension presents a different trend. Unlike the DDI embeddings, increasing the KG embedding dimension does not always lead to performance improvements. While moderate increments (up to 256 dimensions) contribute to slight enhancements in accuracy, recall, and precision, further increasing the dimension beyond this point results in a decline in performance, particularly in the MCC and Pre metrics. This suggests that excessively large KG embeddings may introduce noise or redundant information, which negatively impacts model generalization.

These findings show that selecting an appropriate embedding dimension is crucial for balancing information richness and computational cost. Larger embeddings can capture more complex interaction patterns, but overly large sizes may introduce redundancy and slow down training. Therefore, a suitable embedding size is essential to ensure HEENN achieves strong performance while remaining efficient.

*3) The Impact of Neighborhood Sampling Size on Model Performance:* A large neighborhood sampling size may cause the model to focus on the global structure, neglecting important local information, which can reduce performance. A small sampling size may lose crucial data and limit learning. Thus, choosing an optimal size that balances both local and global information is key.

The figure illustrates that the predicted results exhibit a stronger correlation with the global structure of the dataset. As the neighborhood sampling size increases, model performance improves. However, it is important to account for the computational resources and time needed for training, as larger sampling sizes increase computational complexity. Therefore, identifying the optimal trade-off between capturing local neighborhood information and considering the global structure is critical for maximizing model performance. In our case, as shown in Fig. 8(d), a neighborhood sampling size of 60 provides an effective balance between these two aspects.

To maintain model simplicity and reduce the number of hyperparameters, we applied the same neighborhood sampling size (60) to both the PAN and PPI datasets. For both the PAN and PPI datasets, the same neighborhood sampling size of 60 was applied. In the PAN dataset, 3751 proteins have more than 60 neighbors, while in the PPI dataset, only 80 proteins meet this criterion. This difference in the number of proteins with more than 60 neighbors indicates that, despite the contrasting network densities, a sampling size of 60 enables the HEENN model to achieve optimal performance on both the densely connected PAN dataset and the sparser PPI network.

*4) The Impact of Number of GAT Layers on Model Performance:* Fig. 8(e) demonstrates how varying the number of GAT

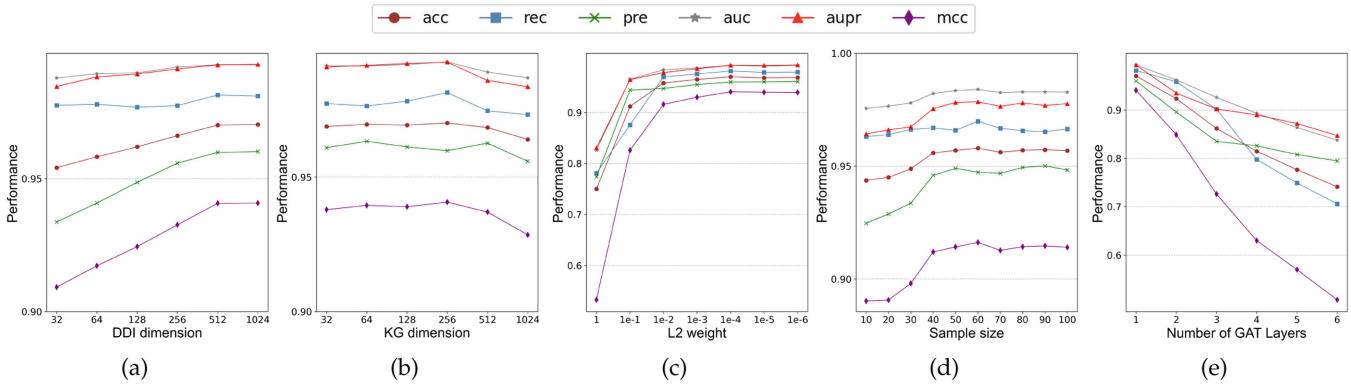


Fig. 8. Parameter sensitivity analysis across different settings.

layers affects the model's predictive performance. When the number of GAT layers is set to 1, the model achieves its highest scores across all evaluation metrics. It indicates strong prediction capability and generalization.

However, as the number of GAT layers increases, the model's performance begins to decline. With two layers, although the results remain competitive, all metrics show noticeable drops compared to the single-layer setting. When extended to five or six layers, the decline becomes more significant. At six layers, the MCC is only 0.5078, and the accuracy drops to 0.7412. This downward trend suggests that deeper GAT architectures introduce over-smoothing, causing node features to become increasingly indistinct and less informative. Additionally, deeper networks may accumulate more noise during feature aggregation, further impairing performance.

Therefore, we choose to use a single GAT layer in our final model. This setting strikes a balance between representation power and stability, and avoids the negative effects of excessive depth.

## VI. Conclusion

In this paper, we propose HEENN, a Hybrid End-to-End Neural Network for protein-protein interaction (PPI) prediction. HEENN integrates two Graph Attention Networks (GATs) to extract local structural and semantic features from the Protein Associated Network (PAN) and PPI network, effectively capturing neighborhood-level relationships. In parallel, an AutoEncoder is employed to model global contextual information, learning high-level abstract representations that complement the localized signals from GATs. To combine these heterogeneous feature types, HEENN adopts an attention-based fusion mechanism that adaptively assigns weights to local and global features, ensuring that the most informative components contribute more to the final representation.

Extensive experiments demonstrate that HEENN outperforms eleven state-of-the-art models across multiple metrics. Its performance stability across different datasets highlights its adaptability to networks with varying densities and topologies. Moreover, HEENN can effectively prioritize candidate interactions, helping biologists focus experimental validation on the most promising targets. To further improve the interpretability of HEENN, we plan to explore attention weight visualization and layer-wise relevance propagation to highlight key nodes and interactions that drive model decisions. Additionally, incorporating domain-specific attribution methods could facilitate biological insight into interaction mechanisms beyond the case study.

In future work, we plan to apply HEENN to disease-specific pathways and integrate self-supervised learning to better utilize unlabeled data. We also aim to explore multimodal graph learning frameworks and investigate granular ball theory to enhance the robustness of input representation. Specifically, we believe granular ball theory can help capture local data density and structural uncertainty in high-dimensional biological networks, potentially leading to more stable and noise-resistant feature representations for PPI prediction. Overall, HEENN offers a scalable, interpretable, and high-performance solution for advancing computational biology tasks such as PPI prediction.

## References

[1] I. R. Humphreys et al., "Computed structures of core eukaryotic protein complexes," *Science*, vol. 374, no. 6573, 2021, Art. no. eabm4805.
[2] C. Chen, Q. Zhang, Q. Ma, and B. Yu, "LightGBM-PPI: Predicting protein-protein interactions through LightGBM with multi-information fusion," *Chemometrics Intell. Lab. Syst.*, vol. 191, pp. 54–64, 2019.
[3] D. Wu, W. Sun, Y. He, Z. Chen, and X. Luo, "MKG-FENN: A multimodal knowledge graph fused end-to-end neural network for accurate drug–drug interaction prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 10216–10224.
[4] X. Li, P. Han, G. Wang, W. Chen, S. Wang, and T. Song, "SDNN-PPI: Self-attention with deep neural network effect on protein-protein interaction prediction," *BMC Genomic.*, vol. 23, no. 1, 2022, Art. no. 474.
[5] K. Ma et al., "PPRTGI: A personalized PageRank graph neural network for TF-target gene interaction detection," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 21, no. 3, pp. 480–491, May/Jun. 2024.
[6] X. Luo, L. Wang, P. Hu, and L. Hu, "Predicting protein-protein interactions using sequence and network information via variational graph autoencoder," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 5, pp. 3182–3194, Sep./Oct. 2023.
[7] F. Zhang, S. Chang, B. Wang, and X. Zhang, "DSSGNN-PPI: A protein–protein interactions prediction model based on double structure and sequence graph neural networks," *Comput. Biol. Med.*, vol. 177, 2024, Art. no. 108669.
[8] J. Hu, Z. Li, B. Rao, M. A. Thafar, and M. Arif, "Improving protein-protein interaction prediction using protein language model and protein network features," *Anal. Biochem.*, vol. 693, 2024, Art. no. 115550.
[9] Z. Zhang, Q. Zhang, J. Xiao, S. Ding, and Z. Li, "MFC-PPI: Protein–protein interaction prediction with multimodal feature fusion and contrastive learning," *J. Supercomputing*, vol. 81, no. 4, pp. 1–19, 2025.
[10] S. Zhao, Z. Cui, G. Zhang, Y. Gong, and L. Su, "MGPPI: Multiscale graph neural networks for explainable protein–protein interaction prediction," *Front. Genet.*, vol. 15, 2024, Art. no. 1440448.

[11] Z. Zhou et al., "ProAffinity-GNN: A novel approach to structure-based protein–protein binding affinity prediction via a curated data set and graph neural networks," *J. Chem. Inf. Model.*, vol. 64, no. 23, pp. 8796–8808, 2024.
[12] C. Wu et al., "PROBind: A web server for prediction, analysis and visualization of protein-protein and protein-nucleic acid binding residues," *bioRxiv*, Feb. 2025, doi: 10.1101/2025.02.08.637237.
[13] J. Yang, Y. Li, G. Wang, Z. Chen, and D. Wu, "An end-to-end knowledge graph fused graph neural network for accurate protein-protein interactions prediction," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 21, no. 6, pp. 2518–2530, Nov./Dec. 2024.
[14] X. Du, S. Sun, C. Hu, Y. Yao, Y. Yan, and Y. Zhang, "DeepPPI: Boosting prediction of protein–protein interactions with deep neural networks," *J. Chem. Inf. Model.*, vol. 57, no. 6, pp. 1499–1510, 2017.
[15] M. Chen et al., "Multifaceted protein–protein interaction prediction based on siamese residual RCNN," *Bioinformatics*, vol. 35, no. 14, pp. i305–i314, 2019.
[16] X. Hu, C. Feng, Y. Zhou, A. Harrison, and M. Chen, "DeepTrio: A ternary prediction system for protein–protein interaction using mask multiple parallel convolutional neural networks," *Bioinformatics*, vol. 38, no. 3, pp. 694–702, 2022.
[17] Y. Yao, X. Du, Y. Diao, and H. Zhu, "An integration of deep learning with feature embedding for protein–protein interaction prediction," *PeerJ*, vol. 7, 2019, Art. no. e7126.
[18] F. Yang, K. Fan, D. Song, and H. Lin, "Graph-based prediction of protein-protein interactions with attributed signed graph embedding," *BMC Bioinf.*, vol. 21, pp. 1–16, 2020.
[19] M. Baranwal et al., "Struct2Graph: A graph attention network for structure based predictions of protein–protein interactions," *BMC Bioinf.*, vol. 23, no. 1, 2022, Art. no. 370.
[20] G. Cheng et al., "SPNet: Siamese-prototype network for few-shot remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5608011.
[21] W. Ma, W. Bao, Y. Cao, B. Yang, and Y. Chen, "Prediction of protein-protein interaction based on deep learning feature representation and random forest," in *Proc. 17th Int. Conf. Intell. Comput. Theories Appl.*, Springer, 2021, pp. 654–662.
[22] S. Sledzieski, R. Singh, L. Cowen, and B. Berger, "Sequence-based prediction of protein-protein interactions: A structure-aware interpretable deep learning model," *BioRxiv*, Jan. 2021, doi: 10.1101/2021.01.22.427866.
[23] Q. Guo, X. Yang, F. Zhang, and T. Xu, "Perturbation-augmented graph convolutional networks: A graph contrastive learning architecture for effective node classification tasks," *Eng. Appl. Artif. Intell.*, vol. 129, 2024, Art. no. 107616.
[24] C. Wang, Y. Wang, P. Ding, S. Li, X. Yu, and B. Yu, "ML-FGAT: Identification of multi-label protein subcellular localization by interpretable graph attention networks and feature-generative adversarial networks," *Comput. Biol. Med.*, vol. 170, 2024, Art. no. 107944.
[25] Q. Guo, X. Yang, M. Li, and Y. Qian, "Collaborative graph neural networks for augmented graphs: A local-to-global perspective," *Pattern Recognit.*, vol. 158, 2025, Art. no. 111020.
[26] Y. Zhang, X. Wang, C. Shi, X. Jiang, and Y. Ye, "Hyperbolic graph attention network," *IEEE Trans. Big Data*, vol. 8, no. 6, pp. 1690–1701, Dec. 2022.
[27] J. Ben, Q. Sun, K. Liu, X. Yang, and F. Zhang, "Multi-head multi-order graph attention networks," *Appl. Intell.*, vol. 54, no. 17, pp. 8092–8107, 2024.
[28] Y. Dong, Q. Liu, B. Du, and L. Zhang, "Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 31, pp. 1559–1572, 2022.
[29] X. Lu, G. Chen, J. Li, X. Hu, and F. Sun, "MAGCN: A multiple attention graph convolution networks for predicting synthetic lethality," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 5, pp. 2681–2689, Sep./Oct. 2023.
[30] X. Mo, Z. Huang, Y. Xing, and C. Lv, "Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9554–9567, Jul. 2022.
[31] Z. Cheng, C. Yan, F.-X. Wu, and J. Wang, "Drug-target interaction prediction using multi-head self-attention and graph attention network," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 19, no. 4, pp. 2208–2218, Jul./Aug. 2022.
[32] Z. He, H. Shao, Z. Ding, H. Jiang, and J. Cheng, "Modified deep autoencoder driven by multisource parameters for fault transfer prognosis of aeroengine," *IEEE Trans. Ind. Electron.*, vol. 69, no. 1, pp. 845–855, Jan. 2022.
[33] T.-H. Wang, X. Hu, H. Jin, Q. Song, X. Han, and Z. Liu, "AutoRec: An automated recommender system," in *Proc. 14th ACM Conf. Recommender Syst.*, 2020, pp. 582–584.
[34] P. Li, Y. Pei, and J. Li, "A comprehensive survey on design and application of autoencoder in deep learning," *Appl. Soft Comput.*, vol. 138, 2023, Art. no. 110176.
[35] T. N. Duong et al., "A novel autorec-based architecture for recommendation system," in *Proc. 10th Int. Conf. Commun. Electron.*, 2024, pp. 469–474.
[36] T. N. Duong, T. T. T. Pham, T. G. Do, T. K. P. Dinh, and M. H. Tran, "A generalized autorec framework applying content-based information for resolving data sparsity problem," in *Proc. 12th Int. Symp. Inf. Commun. Technol.*, 2023, pp. 181–188.
[37] Q. Cui et al., "AutoRec: A comprehensive platform for building effective and explainable recommender models," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. in Databases*, Springer, 2021, pp. 541–545.
[38] C. Zhang, Y. Geng, Z. Han, Y. Liu, H. Fu, and Q. Hu, "Autoencoder in autoencoder networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 2263–2275, Feb. 2024.
[39] X.-R. Su, L. Hu, Z.-H. You, P.-W. Hu, and B.-W. Zhao, "Multi-view heterogeneous molecular network representation learning for protein–protein interaction prediction," *BMC Bioinf.*, vol. 23, no. 1, 2022, Art. no. 234.
[40] D. Szklarczyk et al., "The STRING database in 2025: Protein networks with directionality of regulation," *Nucleic Acids Res.*, vol. 53, no. D1, pp. D730–D737, 2025.
[41] C. Knox et al., "DrugBank 6.0: The DrugBank knowledgebase for 2024," *Nucleic Acids Res.*, vol. 52, no. D1, pp. D1265–D1275, 2024.
[42] H. Zhao et al., "LncTarD 2.0: An updated comprehensive database for experimentally-supported functional lncRNA–target regulations in human diseases," *Nucleic Acids Res.*, vol. 51, no. D1, pp. D199–D207, 2023.
[43] S. Cui et al., "miRTarBase 2025: Updates to the collection of experimentally validated microRNA–target interactions," *Nucleic Acids Res.*, vol. 53, no. D1, pp. D147–D156, 2025.
[44] A. P. Davis et al., "Comparative toxicogenomics database's 20th anniversary: Update 2025," *Nucleic Acids Res.*, vol. 53, no. D1, pp. D1328–D1334, 2025.
[45] Y.-A. Huang, Z.-H. You, X. Chen, K. Chan, and X. Luo, "Sequence-based prediction of protein-protein interactions using weighted sparse representation model combined with global encoding," *BMC Bioinf.*, vol. 17, pp. 1–11, 2016.
[46] T. Sun, B. Zhou, L. Lai, and J. Pei, "Sequence-based prediction of protein protein interaction using a deep-learning algorithm," *BMC Bioinf.*, vol. 18, pp. 1–8, 2017.
[47] S. Hashemifar, B. Neyshabur, A. A. Khan, and J. Xu, "Predicting protein–protein interactions through sequence-based deep learning," *Bioinformatics*, vol. 34, no. 17, pp. i802–i810, 2018.
[48] X.-R. Su, Z.-H. You, L. Hu, Y.-A. Huang, Y. Wang, and H.-C. Yi, "An efficient computational model for large-scale prediction of protein–protein interactions based on accurate and scalable graph embedding," *Front. Genet.*, vol. 12, 2021, Art. no. 635451.
[49] T. Lyu, J. Gao, L. Tian, Z. Li, P. Zhang, and J. Zhang, "MDNN: A multimodal deep neural network for predicting drug-drug interaction events," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 3536–3542.
[50] B. Song, X. Luo, X. Luo, Y. Liu, Z. Niu, and X. Zeng, "Learning spatial structures of proteins improves protein–protein interaction prediction," *Brief. Bioinf.*, vol. 23, no. 2, 2022, Art. no. bbab558.
[51] S. Xie et al., "HNSPPI: A hybrid computational model combing network and sequence information for predicting protein–protein interaction," *Brief. Bioinf.*, vol. 24, no. 5, 2023, Art. no. bbad261.
[52] H. Gao, C. Chen, S. Li, C. Wang, W. Zhou, and B. Yu, "Prediction of protein-protein interactions based on ensemble residual convolutional neural network," *Comput. Biol. Med.*, vol. 152, 2023, Art. no. 106471.